IN THE CLAIMS:

1. (Currently Amended) A method for comparing a first order-independent data set comprising unique elements with a second order-independent data set comprising unique elements, ~~the method~~ comprising ~~the steps of~~:

    (a) for each entry in the first data set, placing the entry in a hash table, wherein the first data set is stored on a source storage system;

    (b) selecting an entry from the second data set, wherein the second data set is located on a destination storage system and the source storage system and the destination storage system are separate stand alone storage systems;

    (c) looking up the selected entry in the hash table;

    (d) removing, in response to locating the selected entry in the hash table, the selected entry from the hash table;

    (e) determining if additional second data set entries exist;

    (f) looping to step (b) in response to identifying additional second data set entries; and

    (g) reporting a difference between the first data set and the second data set in response to at least one first data set entry remaining in the hash table.

2. (Currently Amended) The method of claim 1 further comprising ~~the step of~~ identifying, in response to not locating the selected entry in the hash table, that the selected entry is second data set unique.

3. (Currently Amended) The method of claim 1 further comprising ~~the step of~~ performing, in response to not locating the selected entry in the hash table, a remedial function.

4. (Original) The method of claim 3 wherein the remedial function comprises deleting the selected entry of the second data set.

5. (Currently Amended) The method of claim 1 further comprising ~~the step of~~ identifying in response to no additional entries existing, any remaining entries in the hash table data as being first data set unique.

6. (Currently Amended) The method of claim 1 further comprising ~~the step of~~ performing in response to no additional entries existing, a remedial function.

7. (Original) The method of claim 6 wherein the remedial function comprises deleting the selected entry of the first data set.

8. (Currently Amended) The method of claim 6 wherein the remedial function comprises ~~the step of~~ transferring the selected entry from the first data set to the second data set.

9. (Original) The method of claim 1 wherein the step of removing the selected entry from the hash table occurs in response to identifying a match between a selected entry of the first data set and an entry of the second data set.

10. (Original) The method of claim 1 wherein the hash table comprises a B-tree.

11. (Original) The method of claim 1 wherein the hash table comprises a fast lookup data structure.

12. (Original) The method of claim 1 wherein the first data set comprises a set of directory entries on a source system.

13. (Original) The method of claim 1 wherein the second data set comprises a set of entries of a directory on a destination system.

14. (Original) The method of claim 1 wherein the first data set comprises a set of direc-
tory entries on a destination system.

15. (Original) The method of claim 1 wherein the second data set comprises directory
entries on a source data set.

16. (Previously Presented) The method of claim 1 wherein the first data set and the sec-
ond data set are on different storage devices.

17. (Currently Amended) A system for comparing a first data set with a second data set,
the system comprising:

    (a) means for placing each entry of the first data set in a hash table, wherein the
first data set is stored on a source storage system;

    (b) means for selecting an entry from the second data set, wherein the second data
set is located on a destination storage system and the source storage system and the desti-
nation storage system are separate stand alone storage systems;

    (c) means for looking up the selected entry in the hash table;

    (d) means for removing, in response to locating the selected entry in the hash ta-
ble, the selected entry from the hash table;

    (e) means for determining if additional second data set entries exist;

    (f) means for looping to step (b) in response to identifying additional second data
set entries; and

    (g) means for reporting a difference between the first data set and the second data
set in response to at least one first data set entry remaining in the hash table.

18. (Original) The system of claim 17 wherein the hash table comprises a B-tree.

19. (Currently Amended) A computer readable medium, including program instructions executing on a computer, the program instructions including instructions for performing the steps of:

(a) for each entry in a first data set, placing the entry in a hash table, wherein the first data set is stored on a source storage system;

(b) selecting an entry from a second data set, wherein the second data set is located on a destination storage system and the source storage system and the destination storage system are separate stand alone storage systems;

(c) looking up the selected entry in the hash table;

(d) removing, in response to locating the selected entry in the hash table, the selected entry from the hash table;

(e) determining if additional second data set entries exist; and

(f) looping to step (b) in response to identifying additional second data set entries; and

(g) reporting a difference between the first data set and the second data set in response to at least one first data set entry remaining in the hash table.

20. (Currently Amended) A method for comparing a first data set with a second data set, the method comprising the steps of:

creating a hash table of entries of the first data set, wherein the first data set is stored on a source storage system;

locating, for each entry in the second data set, an entry in the hash table, wherein the second data set is located on a destination storage system and the source storage system and the destination storage system are separate stand alone storage systems;

removing, in response to locating an entry in the hash table, the located entry; and

recording, in response to at least one entry remaining in the hash table, a difference between the first data set and the second data set.

1   21. (Currently Amended) A method for comparing a first data set with a second data set,

2   the method comprising the steps of:

3   creating a hash table of entries of the first data set, wherein the first data set is

4   stored on a source storage system;

5   locating, for each entry in the second data set, an entry in the hash table, wherein

6   the second data set is located on a destination storage system and the source storage sys-

7   tem and the destination storage system are separate stand alone storage systems;

8   removing, in response to locating an entry in the hash table, the located entry; and

9   recording, in response to not locating an entry in the hash table, that the entry in

10   the second data set is second date set unique; and

11   reporting a difference between the first data set and the second data set in re-

12   sponse to at least one first data set entry remaining in the hash table.

1   22. (Currently Amended) A method for comparing a first data set with a second data set,

2   the method comprising the steps of:

3   (a) selecting an entry from the first data set, wherein the first data set is stored on

4   a source storage system;

5   (b) determining if the selected entry from the first data set is in a hash table;

6   (c) adding, in response to determining that the selected entry from the first data

7   set is not in the hash table, the selected entry from the first data set to the hash table;

8   (d) removing from the hash table, in response to determining that the selected en-

9   try from the first data set is in the hash table, the selected entry from the first data set;

10   (e) selecting an entry from the second data set, wherein the second data set is lo-

11   cated on a destination storage system and the source storage system and the destination

12   storage system are separate stand alone storage systems;

13   (f) determining if the selected entry from the second data set is in the hash table;

14   (g) adding, in response to determining that the selected entry from the second data

15   set is not in the hash table, the selected entry from the second data set to the hash table;

16   (h) removing, in response to determining that the selected entry from the second

17   data set is in the hash table, the selected entry from the second data set from the hash ta-

18   ble;

19   (i) independently continuing steps (a) through (d) and (e) through (h) for all en-

20   tries in the first and second data sets until both the first and second data sets have been

21   completely processed; and

22   (j) reporting a difference between the first data set and the second data set in re-

23   sponse to at least one entry remaining in the hash table.

1   23. (Currently Amended) The method of claim 22 wherein the step of adding the selected

2   entry from the first data set to the hash table further comprises ~~the step of~~ including in-

3   formation with the selected entry from the first data set identifying the selected entry

4   from the first data set as originating from the first data set.

1   24. (Currently Amended) The method of claim 22 wherein the step of adding the selected

2   entry from the second data set to the hash table further comprises ~~the step of~~ including

3   information with the selected entry from the second data set identifying the selected entry

4   from the second data set as originating from the second data set.

1   25. (Previously Presented) The method of claim 22 wherein the step of removing the se-

2   lected entry from the second data set from the hash table occurs in response to identifying

3   a match between a selected entry from the second data set and an entry from the first data

4   set.

1   26. (Currently Amended) The method of claim 22 further comprising ~~the step of~~:

2   (k) recording all entries remaining in the hash table as being unique to either the

3   first data set or the second data set.

1   27. (Original) The method of claim 22 wherein the hash table comprises a B-tree.

1 28. (Original) The method of claim 22 wherein the hash table comprises a fast lookup

2 data structure.

1 29. (Original) The method of claim 22 wherein the first data set comprises a set of direc-

2 tory entries on a source system.

1 30. (Original) The method of claim 22 wherein the second data set comprises a set of di-

2 rectory entries on a destination system.

1 31. (Original) The method of claim 22 wherein the first data set and second data set are

2 on different storage devices.

1 32. (Currently Amended) A system for performing a consistency check of a source direc-

2 tory replicated to a destination directory by comparing entries in the source and destina-

3 tion directories, the system comprising:

4   one or more storage disks ~~adapted~~ configured to store one or more of a group con-

5 sisting of the source directory and the destination directory; and

6   a process ~~adapted~~ configured to compare entries in the source directory with en-

7 tries in the destination directory by walking the source and destination directories only

8 once, whereby utilization of storage subsystems associated with the source and destina-

9 tion directories is limited by only walking each of the source and destination directories

10 once, and further ~~adapted~~ configured to report a difference between the source directory

11 and the destination directory, wherein the source directory is located on a source storage

12 system and the destination directory is located on a destination storage system and the

13 source storage system and the destination storage system are separate stand alone storage

14 systems.

33. (Original) The system of claim 32 wherein the process executes on a computer asso-
ciated with the source directory.

34. (Original) The system of claim 32 wherein the process executes on a computer asso-
ciated with the destination directory.

35. (Original) The system of claim 32 wherein the process is further adapted to remove
matching entries from a hash table, whereby future look up operations in the hash table
are enabled to be performed faster due to a smaller size of the hash table.

36. (Currently Amended) A system for performing a consistency check of a source direc-
tory and a destination directory by comparing entries in the source and destination direc-
tories, the system comprising:
    a processor ~~adapted~~ configured to select alternating entries from the source and
destination directories to be added to a hash table and further adapted to remove matching
entries from the hash table, whereby a size of the hash table is limited to a number of dis-
similar entries of the source and destination directories, and further ~~adapted~~ configured to
report a difference between the source directory and the destination directory in response
to the number of dissimilar entries being greater than zero, wherein the source directory
is located on a source storage system and the destination directory is located on a destina-
tion storage system and the source storage system and the destination storage system are
separate stand alone storage systems.

37. (Currently Amended) A system for comparing entries in a source directory with en-
tries on a destination directory to ensure consistency of replicated data between the
source and destination directories, the system comprising:
    a computer associated with at least one of the source and destination directories,
the computer comprising a directory comparison process ~~adapted~~ configured to perform a
comparison of entries in the source and destination directories by walking each directory

9

7   once and placing entries in a hash table and further ~~adapted~~ configured to remove match-

8   ing entries from the hash table, whereby computational cost is reduced for future look up

9   operations in the hash table, wherein the source directory is located on a source storage

10   system and the destination directory is located on a destination storage system and the

11   source storage system and the destination storage system are separate stand alone storage

12   systems.

1   38. (Currently Amended) The system of claim 37 wherein the directory comparison proc-

2   ess is further ~~adapted~~ configured to alternate in selecting entries from the source and des-

3   tination directories when walking the source and destination directories.

1   39. (Previously Presented) The method of claim 1 wherein the step of reporting com-

2   prises recording the difference on a disk.

1   40. (Previously Presented) The method of claim 22 wherein the step of reporting com-

2   prises recording the difference on a disk.

1   41. (Previously Presented) The system of claim 32 wherein the process is further adapted

2   to report the difference by recording the difference on the storage disks.